TP 2 : Densité Spectrale de Puissance

1 Objectifs et évaluation

L'objectif de ce TP est triple

1. Insérer le filtrage adapté dans la chaîne de traitements du TP précédent
2. analyser l'impact du filtre de mise en forme sur la Densité Spectrale de Puissance (DSP) des signaux générés par les communications numériques
3. mettre en œvre la chaîne de réception adaptée à la présence du filtre de mise en forme.

Afin de répondre à ces objectifs, vous écrirez un script interactif de Matlab (notebook) ainsi que des fonctions Matlab.

Vos codes et votre script interactif sont à rendre en fin de séance sur l'interface Thor `https://thor.enseirb-matmeca.fr/ruby/`.

Pour ce deuxième TP, les fonctions du premier TP vous sont fournies au format .p . Vous pouvez faire appel à ces fonctions si jamais vous n'avez pas réussi à développer toutes les fonctions du premier TP.

2 Filtrage de mise en forme

L'objectif de cette partie est de réaliser une simulation le filtrage de mise en forme d'un signal modulé.

Dans le TP1, le signal contenant les symboles S_m est, par définition, échantillonné à période T_s. Pour la suite nous supposerons que $T_s = 1\mu s$.

À temps continu, le signal à la sortie du filtre de mise en forme, s'écrit :

$$s_l(t) = \sum_{m \in \mathbf{Z}} S_m g(t - mT_s) \tag{1}$$

Nous souhaitons échantillonner ce signal à la fréquence $F_e = \frac{1}{T_e} = 10MHz$. Il est aisé de remarquer que

$$\begin{aligned} s_p &= s_l(pT_e) \qquad (2) \\ &= \sum_{m \in \mathbf{Z}} S_m g(pT_e - mT_s) \\ &= \sum_{m \in \mathbf{Z}} S_m g((p - m\frac{T_s}{T_e})T_e) \\ &= \sum_{m \in \mathbf{Z}} S_m g_{p - m\frac{T_s}{T_e}} \qquad (3) \end{aligned}$$

où $g_n = g(nT_e)$ représente le filtre de mise en forme échantillonné avec la période F_e. On remarque de plus que l'équation (3) fait apparaître le ratio des deux périodes d'échantillonnage $F_{se} = \frac{T_s}{T_e}$. Par la suite, ce ratio sera appelé "facteur de sur-échantillonnage", avec les paramètres que nous avons choisis, $F = 10$.

Pour pouvoir implémenter l'équation (3) en Matlab, il suffit de remarquer qu'elle peut être réécrite comme suit

$$s_p = \left(\sum_{m \in \mathbf{Z}} S_m \delta_{p-mF_{se}} \right) * g_p \tag{4}$$

L'opération $S_m \rightarrow \sum_{m \in \mathbf{Z}} S_m \delta_{p-mF_{se}}$ s'appelle **sur-échantillonnage (upsampling en anglais** consiste à créer un vecteur F_{se} fois plus grand que le vecteur contenant les symboles construit comme suit

$$S_{se} = [S_0, 0, \ldots, 0 S_1, 0, \ldots, 0 \ldots S_{N_s-1}, 0, \ldots, 0]. \tag{5}$$

La fonction Matlab qui effectuera la mise en forme s'appellera `shaping`. Cette fonction construit le **signal complexe en bande de base** (nommé $s_l(t)$ dans le cours) échantillonné à la fréquence F_e une à partir des symboles modulés contenus dans le vecteur `s`, du facteur F_{se} et du filtre $g(t)$ échantillonné à la fréquence F_e. `shaping` aura le prototype donné ci-dessous.

Listing 1 – Fonction **shaping.m**.

```
function sl = shaping(s, Fse, g)
% s : (vecteur complexe) vecteur des symboles
% Fse: (int) Fréquence de sur-échantillonnage
% g: (vecteur complexe) filtre de mise en forme
   échantillonné
```

Dans votre notebook : Faites appel à la fonction `shaping` pour générer un signal correspondant à l'envoi de $N_s = 10$ symboles BPSK ($S_n \in \{\pm 1\}$) Vous considérerez, **dans un premier temps**, un filtre de mise en forme rectangulaire défini par :

$$g(t) = \begin{cases} 1 & \text{si} \quad 0 \leq t < T_s \\ \quad 0 & \text{sinon} \end{cases} \tag{6}$$

Afficher le signal $s_l(t)$ et annoter votre graphique pour montrer l'effet du "retard de groupe" pour chaque symbole.

3 Densité spectrale de puissance

On s'intéresse dans cette partie à l'étude de la densité spectrale de puissance du signal $s_l(t)$. La DSP théorique d'un signal de communications numériques peut être calculée à l'aide de la formule suivante (cas particulier de la formule de Bennett)

$$\Gamma_{s_l}(f) = \frac{\sigma_a^2}{T_s} |G(f)|^2 \tag{7}$$

où σ_a^2 est la variance des symboles

$$\sigma_a^2 = \frac{1}{M} \sum_{m=0}^{M-1} |s_m|^2 \tag{8}$$

T_s est le temps symbole et $G(f)$ la réponse en fréquence du filtre de mise en forme.

Le calcul des DSP expérimentales $\hat{\Gamma}_{sl}(f)$ se fera ici en utilisant la méthode du périodogramme moyenné. Pour un signal x_n échantillonné à la fréquence F_e, le périodogramme moyenné pour des fenêtres de N échantillons s'écrit de la façon suivante

$$\hat{\Gamma}_x(f) = \frac{Fe}{PN} \sum_{p=0}^{P-1} \left| \sum_{n=0}^{N-1} x_{pN+n} e^{-j2\pi n \frac{f}{F_e}} \right|^2 . \tag{9}$$

D'un point de vue algorithmique, les étapes pour calculer le périodogramme moyenné sont les suivantes :

1. Découper le signal `x` de la section précédente en fenêtres de N échantillons. **Vous utiliserez la fonction `reshape` pour celà**,
2. Calculer la FFT de chacune des fenêtres,
3. Calculer le $||^2$ de chacune des FFT calculées,
4. Pour chaque point de fréquence, calculer la moyenne des valeurs obtenues.
5. Multiplier le spectre moyen obtenu par $\frac{F_e}{N}$.

Implémenter le périodogramme moyenné dans une fonction ayant le prototype ci-dessous.

Listing 2 – Fonction **periodogramme_moy.m**.

```
function gamma_sl = periodogramme_moy(x, N, N_FFT, Fe)
% x : (vecteur complexe) signal
% N: (int) Taille de la fenêtre
% N_FFT: (int) nombre de points pour le calcul de la FFT
% Fe : (int) Fréquence d'échantillonnage
```

Dans votre notebook : Tracer sur un même graphique les courbes suivantes :

1. Périodogramme moyenné du signal s_l pour $N_s = 5000$ symboles QPSK mis en formes par un filtre rectangulaire. Pour le périodogramme, vous utiliserez les paramètres suivants : $N = N_{FFT} = 512$.
2. Périodogramme moyenné du signal s_l, pour $N_s = 5000$ symboles 8-PSK mis en formes par un filtre rectangulaire, toujours en considérant $N = N_{FFT} = 512$.
3. Périodogramme moyenné du signal s_l, pour $N_s = 5000$ symboles 16-PSK mis en formes par un filtre rectangulaire, toujours en considérant $N = N_{FFT} = 512$.
4. DSP théorique $\Gamma_{sl}(f)$ obtenue en TD.

Commenter votre résultat.

4 Récepteur

Écrire la fonction `recepteur_optimal` qui implémentera un récepteur optimal. Un récepteur est dit optimal s'il vérifie les conditions suivantes :

1. son filtre est adapté au filtre d'émission (et filtre du canal s'il y en a un)
2. la cascade filtre de mise en forme - filtre adapté vérifie le critère de Nyquist.

La fonction `recepteur_optimal` aura le prototype suivant :

Listing 3 – Fonction **recepteur_optimal**.

```
function r_n = recepteur_optimal(y_l, Fse, g, Ns)
% Arguments :
% y_l : (vecteur complexe) signal reçu (sur-échantillonné)
% Fse : (int) facteur de sur-échantillonnage
% g : (vecteur réel) Réponse impulsionnelle du filtre de
  mise en forme
% Ns : (int) Nombre d'échantillons à renvoyer
% Retour :
% r_n : (vecteur complexe) signal après filtrage adapté et
  échantillonnage à T_s
```

Cette fonction réalisera les actions suivantes sur le signal en sortie de canal y_l (ici $y_l = s_l$) :

— filtrer le signal y_l par le filtre adapté (ce qui donne r_l),

— sous-échantillonner le signal r_l d'un facteur F pour récupérer N_s échantillons (chacun servant à décider un symbole transmis),

Dans votre notebook : Montrez qu'il n'y a pas d'interférences entre symboles en affichant la constellation de r_n après l'envoi de $N_s = 5000$ symboles QPSK. Commentez le diagramme de constellation obtenu.

5 Filtre en Racine de Cosinus Surélevé

Pour cette partie, vous étudierez l'impact d'un changement de filtre de mise en forme. Le fitre porte sera donc remplacé par un filtre en racine de cosinus sur-élevé de roll-off $\alpha = 0.5$ et de temps de propagation de groupe $T_g = 4T_s$.

Dans ses fonctions, Matlab considère le "span" du filtre qui est défini comme $2T_g$.

Vous commenterez les résultats obtenus un bilan comparatif des deux filtres. Ce bilan inclura la comparaison des points suivants :

— **délai sur la réception du premier symbole**

— **bande passante** (obtenue en mesurant la largeur du lobe principal)

— **efficacité spectrale** : obtenue par la formule suivante $\eta = \frac{D_b}{B}$ où D_b est le débit binaire et B la bande passante.

6 Contacts

— Dominique Dallet - dominique.dallet@ims-bordeaux.fr

— Romain Tajan - romain.tajan@ims-bordeaux.fr